

Original citation:

Yang, Xinan and Strauss, Arne. (2017) An approximate dynamic programming approach to attended home delivery management. European Journal of Operational Research .

Permanent WRAP URL:

<http://wrap.warwick.ac.uk/89254>

Copyright and reuse:

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Publisher's statement:

© 2017, Elsevier. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International <http://creativecommons.org/licenses/by-nc-nd/4.0/>

A note on versions:

The version presented here may differ from the published version or, version of record, if you wish to cite this item you are advised to consult the publisher's version. Please see the 'permanent WRAP url' above for details on accessing the published version and note that access may require a subscription.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk

An Approximate Dynamic Programming Approach to Attended Home Delivery Management

Xinan Yang^a, Arne K. Strauss^{b,*}

^a*Dept. of Mathematics, University of Essex, Colchester CO4 3SQ, United Kingdom*

^b*Warwick Business School, University of Warwick, Coventry CV4 7AL, United Kingdom*

Abstract

We propose a new method of controlling demand through delivery time slot pricing in attended home delivery management with a focus on developing an approach suitable for industry-scale implementation. To this end, we exploit a relatively simple yet effective way of approximating delivery costs by decomposing the overall delivery problem into a collection of smaller, area-specific problems. These cost estimations serve as inputs into an approximate dynamic programming method that provides estimates of the opportunity cost associated with having a customer from a specific area book delivery in a specific time slot. These estimates depend on the area and on the delivery time slot under consideration.

Using real, large-scale industry data, we estimate a demand model including a multinomial logit model of customers' delivery time slot choice, and show in simulation studies that we can improve profits by over two per cent in all tested instances relative to using a fixed-price policy commonly encountered in e-commerce. These improvements are achieved despite making strong assumptions in estimating delivery cost. These assumptions allow us to reduce computational run-time to a level suitable for real-time decision making on delivery time slot feasibility and pricing. Our approach provides quantitative insight into the importance of incorporating expected future order displacement costs into opportunity cost estimations alongside marginal delivery costs.

Keywords: E-commerce, Revenue management, Dynamic programming

*Corresponding author

Email addresses: xyangk@essex.ac.uk (Xinan Yang), arne.strauss@wbs.ac.uk (Arne K. Strauss)

1. Introduction

Online grocery sales are experiencing double digit growth in the United Kingdom (12.5 per cent in 2015), and new market entrants like Amazon are increasing competitive pressure on the incumbents to maintain their share of the market as reported by the market research company Mintel (2016). Fulfilment logistics are one of the main cost drivers in the business; hence it is important for retailers to balance carefully the need for high customer service levels in terms of narrow home delivery time slots with the associated costs of service provision. Mintel (2016) find that 32 per cent of current online grocery shoppers shop online because of improvements in delivery time slots. Most UK retailers have moved to one-hour slots offered over a wide time span, from around 6am to 11pm. Whilst customers expect convenient delivery slots, they are not prepared to pay much for peak-time delivery; between £6 and £7 is the maximum currently charged by UK retailers. However, Yang et al. (2016) show in an empirical study that even small fee differences may influence customers' delivery time slot decisions, and may lead to overall improved profitability. Mintel (2016) further find that about 80 per cent of online grocery customers in the UK pay per delivery, whilst the remaining 20 per cent have delivery passes. Therefore, influencing customers' delivery time slot choices through dynamic pricing seems a promising way of achieving more profitable delivery schedules.

This problem has recently received considerable attention from the academic community (see Section 2), and various contributions in this context have been made toward estimating choice behavior, dynamic pricing, controlling the availability of delivery time slots, and approximating the delivery costs associated with a set of orders.

In this work, we develop a dynamic delivery slot pricing policy to manage demand over a finite booking horizon prior to the actual delivery date such that we maximize expected profit. We do not consider same-day delivery. The policy is based on a customer delivery slot choice model and reflects in its pricing the so-called opportunity cost associated with each delivery slot option. The opportunity cost of a delivery slot may be interpreted as its value, which is influenced by the anticipated displaced order cost (meaning that some future orders are lost due to capacity constraints if the customer selects this time slot) and by marginal delivery costs.

The main challenge is to calculate the opportunity costs; once these are known, we only

need a fast method to evaluate the feasibility of a delivery in a given time slot and area, and to optimize delivery time slot prices when a customer arrives wanting to book a delivery. The opportunity cost calculation and feasibility check both require a way of approximating the NP-hard, capacitated vehicle routing problem with time windows. We decompose this problem into a collection of smaller, independent subproblems corresponding to clusters of postcodes, and use a continuous approximation of the total traveling distance. Based on this decomposition, we propose an approximate dynamic programming model to estimate the opportunity costs. The latter are used as inputs into the real-time optimization of delivery time slot prices.

The main contribution of our work to the existing literature is to quantify the impact on profit of incorporating expected order displacement costs (in addition to marginal delivery costs) into the opportunity cost estimate in a way suitable for large-scale applications. Real-time decision making is achieved by using a simple approximation of the vehicle routing problem that allows an easy and quick time slot feasibility check, and by exploiting results available in the literature on how to solve the pricing problem efficiently. We demonstrate the effectiveness of this method in a simulation study based on real data from our industry partner, and show that our approach may produce significant profit improvements against various benchmarks.

From a business perspective, several managerial insights can be gleaned from our numerical experiments. First, dynamic pricing does not necessarily always improve on fixed-price strategies; its success hinges on good opportunity cost estimates that include both marginal delivery costs and expected order displacement cost. Second, it may be better to offer low delivery charges in remote areas with low demand density, even though the cost of deliveries is higher than in areas with high demand density. Our proposed method sets delivery charges in this way in order to stimulate demand in areas with low demand, in contrast to alternative methods (e.g. Yang et al. (2016)) that make delivery charge decisions based solely on delivery cost estimates. In practice, we expect this stimulation to be further reinforced by the so-called neighborhood effect (our study does not take this into account): direct social interaction between neighbors or observation of deliveries to neighbors typically leads to increased demand, as shown by Bell and Song (2007). If charges are based exclusively on estimated marginal delivery costs for a given order, the resulting higher charges in areas with low demand density

are a hindrance to demand growth. Furthermore, such pricing policies give rise to equity concerns regarding systematically disadvantaged customers in remote areas (see Lang et al. (2017)).

This paper is organized as follows: in Section 2 we review and discuss the related literature and in Section 3 we formally state the problem formulation. In Sections 4 and 5 respectively, we then explain how we approximate the delivery costs and value function. Based on these approximations, we obtain a pricing policy, as defined in Section 6. This policy is tested against fixed-price benchmarks in a numerical study presented in Section 7, and we draw conclusions in Section 8.

2. Literature review

For a review of e-fulfillment from an operational research perspective, see Agatz et al. (2008). A recent overview of fulfillment and distribution from a qualitative point of view is provided by Hübner et al. (2016).

In their seminal work, Campbell and Savelsbergh (2005) investigate a dynamic routing and scheduling problem of a grocery vendor who needs to decide which deliveries to accept or reject, and in which time slot to deliver the accepted orders. Customers are assumed to have a certain time slot profile which represents all slots that they are willing to accept; if the order is accepted, the firm assigns one of these slots to the order.

Campbell and Savelsbergh (2005) model demand as an arrival process that is not influenced by the firm’s decisions. In the follow-up paper Campbell and Savelsbergh (2006), they use a relatively simple model of customer behavior to capture the effect of incentives (such as delivery charges) on the probability of a particular time slot being chosen. Their focus is on influencing delivery time slot choices to reduce delivery costs, as opposed to improving total expected profits, which is the focus of our paper.

A more sophisticated model of customer choice, namely multinomial logit (MNL), is used by Asdemir et al. (2009) for dynamic time slot pricing. They propose a dynamic programming (DP) formulation under the assumption that the problem can be addressed independently for each geographical area (e.g. a postcode), and that delivery capacity levels are committed a priori to each delivery time slot. Therefore, delivery costs are fixed, and the objective is to maximize the expected profit from orders. The state space of their DP grows exponentially

in the number of delivery time slots, which makes practical application difficult. In our work, we likewise use the MNL choice model and consider a DP formulation decomposed by geographical area, but we also discuss how to obtain these areas, and how to approximately solve the DP for industry-sized problems. Furthermore, the delivery cost approximation in our model is dynamic, not fixed as in Asdemir et al. (2009).

Agatz et al. (2011) address the geographical dependence issue of the attended home delivery problem. Their approach relates to ours in that they also use the work of Daganzo (1987) to obtain a continuous delivery cost approximation. However, they consider the problem of which delivery time slots to offer in which area so as to reduce delivery costs while meeting service requirements. Their work does not address the problem of how to influence customer choice behavior so as to improve expected profit, which is the aim of our work.

Ehmke and Campbell (2014) examine an integrated routing and scheduling problem in the context of attended home delivery. Their objective is to maximize the number of requests accepted for delivery, subject to retaining feasible tours. Customers' delivery slot choices are assumed to be independent of the firm's decision making, which is limited to accepting or rejecting delivery slot booking requests. In contrast, our study aims to maximize total profit by deciding on delivery time slot charges, which directly influence our model of customers' delivery time slot choice behavior. Similarly, Cleophas and Ehmke (2014) discuss decision making in terms of acceptance or rejection of delivery requests, but also propose to reserve transport capacity for specific delivery areas and time windows with a high expected order value.

Yang et al. (2016) estimate an MNL choice model from real e-grocer data and demonstrate numerically that using this model for time slot pricing to influence demand may improve overall profitability. They employ insertion heuristics to update a pool of feasible routes as orders come in over the booking horizon, and derive marginal delivery cost estimates used as estimates of the opportunity cost of accepting an order into a particular time slot. In our work, we draw on their choice model but use a different (and computationally much more efficient) way of estimating marginal delivery costs. Furthermore, our opportunity cost estimates are not based solely on delivery costs, but also take potential future order displacement costs into account. The work of Cleophas and Ehmke (2014) relates to that of Yang et al. (2016), in that both papers combine demand fulfillment and revenue management. However, the latter

is concerned with time slot pricing and incorporates customer choice modeling, in contrast to the static demand model of Cleophas and Ehmke (2014).

A key difficulty of the attended home delivery problem is estimating routing costs before all (or even any) orders are known. Bühler et al. (2016) propose several linear mixed-integer programs to approximate delivery costs based on a fixed pool of potential routes. Klein et al. (2016) integrate a linear mixed-integer program (MIP) formulation into the dynamic pricing approach of Yang et al. (2016) in attempting to anticipate future customer requests. This approach aims to obtain opportunity cost estimates that feature both delivery costs and revenue implications similar to those we produce in this work. In other words, their MIP formulation can be seen as an approximation of the value function. However, the MIP as proposed by Klein et al. (2016) suffers from computational challenges for industry-sized problems because the number of decision variables grows exponentially in the number of delivery time slots.

Klein et al. (2015) is related to our work in that they consider time slot pricing in attended home delivery under a model of customer choice. Their objective and problem setting are very similar to ours, but they tackle the problem with a different delivery cost estimation, a different choice model, and a different approximation of the value function (namely using an MIP formulation).

3. Problem statement

We consider an e-grocer receiving orders via an online booking system which requires customers to book their delivery when completing their purchase. Orders for a specific delivery day can be received over a finite time horizon until a certain cut-off time, after which no further orders are accepted. Deliveries are made after the cut-off time. We model this as a discrete booking horizon, starting at time $t = 1$ and ending in time period $t = T$. Each time period is sufficiently small for the probability of more than one customer arrival to be negligible. Customers arrive over this time horizon to book their delivery for this specific delivery day. This follows the demand model of Yang et al. (2016) in that we do not consider delivery time slot choice beyond a single day. Customer arrivals follow a homogeneous Poisson process, λ . Note that this homogeneity can be achieved by appropriately defining a non-homogeneous time grid, as explained by Yang et al. (2016). An arriving customer requests delivery in area

$a \in \{1, 2, \dots, n\}$ with probability μ_a , where n is the total number of potential delivery areas. With a probability of σ_{ai} , the number of totes in an order from area a is $i \in I_a$, where I_a is a finite set of order sizes that may be encountered in area a (obtained from historical data). By definition, $\sum_{i \in I_a} \sigma_{ai} = 1, \forall a$. We assume that order size is independent of both time of order placement and delivery charge.

When a customer arrives, we need to check which time slots $s \in \{1, \dots, m\}$ out of the total of m slots are feasible for the desired area given an order size of i (measured in the number of required transport totes). We denote the resulting feasible set by $F_{ai} \subset \{1, \dots, m\}, \forall a, i \in I_a$. We then need to decide which delivery charges to impose on the feasible slots. Faced with the resulting set of feasible slot alternatives $s \in F_{ai}$ for area a and given order size i as well as delivery charges $\vec{d}_a := (d_{as})_{s \in F_{ai}}$, the customer decides according to some choice model when (or whether) to book delivery. This choice model specifies the probability $P_{s, F_{ai}}(\vec{d}_a)$ that a customer will choose slot s , given the vector of delivery charges \vec{d}_a for area a over all feasible slots F_{ai} . If the customer books slot s , we receive a delivery charge of d_{as} and a revenue of ir , where i is the number of totes ordered and r is the average profit per tote.

This control problem can be modeled as a Markov decision process over states labeled (\vec{x}_t, \vec{y}_t) defined for discrete time periods (stages) $t = 1, \dots, T$ which collectively represent the booking horizon. Vector \vec{x}_t consists of components x_{tas} which represent the number of orders accepted in time slot s for area a until time t in the booking horizon. Vector \vec{y}_t consists of components y_{ta} representing the total number of totes collected over all time slots s for area a until time t . As we assume that a van is only loaded once in the morning, y_{ta} is independent of delivery time slots s . Note that this state definition does not capture which order has requested which totes; rather, we only have the aggregated required number of totes y_{ta} . We use this reduced state because it contains sufficient information for delivery cost approximation, as discussed in the next section. Our actions are the setting of delivery time slot charges $\vec{d}_a \in \mathbb{R}_+^{|F_{ai}|}$ for all areas a and order sizes $i \in I_a$. Having taken an action in time period t , we then transition stochastically from state (\vec{x}_t, \vec{y}_t) to state $(\vec{x}_{t+1}, \vec{y}_{t+1})$ according to the distributions of customer arrival λ , order size σ and slot choice $P_{s, F_{ai}}(\vec{d}_a)$. More specifically, if we receive in booking period t an order of i totes for delivery in slot s , then we transition from state (\vec{x}_t, \vec{y}_t) to state $(\vec{x}_t + \mathbf{1}_{as}, \vec{y}_t + i\mathbf{1}_a)$, where $\mathbf{1}_{as}$ ($\mathbf{1}_a$) is the unit vector with 1 in the (a, s) th (a th) position. Our objective is to maximize expected profit over

booking horizon $[1, T]$. In the following, we omit time index t in (\vec{x}_t, \vec{y}_t) since this will be clear from the context of the dynamic programming recursion, which also features time index t .

Let $V_t(\vec{x}, \vec{y})$ denote the value function at period t and state (\vec{x}, \vec{y}) ; this represents the maximum expected profit obtainable from the sales process from time t until cut-off time T . The dynamic programming recursion at stage $t \in \{1, 2, \dots, T\}$ is thus:

$$\begin{aligned}
V_t(\vec{x}, \vec{y}) &= \max_{\vec{d}} \lambda \sum_{a,i} \mu_a \sigma_{ai} \sum_{s \in F_{ai}} P_{s, F_{ai}}(\vec{d}_a) \left[ir + d_{as} + V_{t+1}(\vec{x} + \mathbf{1}_{as}, \vec{y} + i\mathbf{1}_a) \right] + \\
&\quad \left[1 - \lambda \sum_{a,i} \mu_a \sigma_{ai} \sum_{s \in F_{ai}} P_{s, F_{ai}}(\vec{d}_a) \right] V_{t+1}(\vec{x}, \vec{y}) \\
&= \max_{\vec{d}} \lambda \sum_{a,i} \mu_a \sigma_{ai} \sum_{s \in F_{ai}} P_{s, F_{ai}}(\vec{d}_a) \left[ir + d_{as} - \left(V_{t+1}(\vec{x}, \vec{y}) - V_{t+1}(\vec{x} + \mathbf{1}_{as}, \vec{y} + i\mathbf{1}_a) \right) \right] \\
&\quad + V_{t+1}(\vec{x}, \vec{y}) \quad \forall (\vec{x}, \vec{y}) \in \mathcal{X},
\end{aligned} \tag{3.1}$$

where $\mathcal{X} := \{(\vec{x}, \vec{y}) \in \mathbb{N}^{nm} \times \mathbb{N}^n : 0 \leq x_{as} \leq \bar{x} \forall a, s; 0 \leq y_a \leq \bar{y} \forall a\}$ denotes the set of all potential states, and \bar{x} (\bar{y}) is some upper bound on the number of accepted orders (number of required transport totes). In each time period t of the booking horizon, we need to decide on the optimal delivery charges \vec{d} . A customer arriving in this time period with probability λ , is interested in delivery to area a with probability μ_a , orders i totes with probability σ_{ai} and then chooses delivery slot s with probability $P_{s, F_{ai}}(\vec{d}_a)$. In this case, we receive ir profit from the order, plus d_{as} from the delivery charge, and we transition in the next stage ($t+1$) to a new state $(\vec{x} + \mathbf{1}_{as}, \vec{y} + i\mathbf{1}_a)$. If no order is placed, we remain in state (\vec{x}, \vec{y}) .

We let $C(\vec{x}, \vec{y})$ represent an oracle that returns an approximation of the minimum cost for the underlying vehicle routing problem with time windows for the set of orders (\vec{x}, \vec{y}) given a fixed fleet of vehicles with known capacities. If there is no feasible solution for a given state, then $C(\vec{x}, \vec{y}) := \infty$. We denote by $F_{ai}(\vec{x}, \vec{y}) := \{s : C(\vec{x} + \mathbf{1}_{as}, \vec{y} + i\mathbf{1}_a) < \infty\}$ all feasible time slots for area a into which an order with i totes can be feasibly inserted given that we are in state (\vec{x}, \vec{y}) . For brevity of notation, we use the short-hand notation F_{ai} mentioned above.

The value function after cut-off is

$$V_{T+1}(\vec{x}, \vec{y}) = -C(\vec{x}, \vec{y}) \quad \forall (\vec{x}, \vec{y}) \in \mathcal{X}. \tag{3.2}$$

If we can somehow solve this dynamic program (or at least approximate the value function), then we can use the value function in a decision policy in the following form. Given

the arrival of a customer from a known area a with order size i during the booking horizon, we simply need to solve the so-called online decision problem:

$$\{d_{as}^* | s \in F_{ai}\} = \operatorname{argmax}_{s \in F_{ai}} \sum P_{s, F_{ai}}(\vec{d}_a) \left[ir + d_{as} - \left(V_{t+1}(\vec{x}, \vec{y}) - V_{t+1}(\vec{x} + \mathbf{1}_{as}, \vec{y} + i\mathbf{1}_a) \right) \right].$$

The term $\left(V_{t+1}(\vec{x}, \vec{y}) - V_{t+1}(\vec{x} + \mathbf{1}_{as}, \vec{y} + i\mathbf{1}_a) \right)$ represents the opportunity cost of having a customer from area a book delivery in slot s at time t with an order of size i , given that we currently have orders (\vec{x}, \vec{y}) on the books. This opportunity cost can be interpreted as the expected displacement of profits from future orders as a result of accepting this order, so in the revenue management literature, it is sometimes also referred to as the displacement cost. Depending on the choice model, the online decision problem can be solved efficiently, as discussed in Section 6, provided that we have an approximation of the opportunity cost and a way of determining the feasible set of slots F_{ai} . Note that the opportunity cost reflects both the revenue and cost implications of having a customer book a delivery. Yang et al. (2016) approximate the opportunity cost using only the estimated delivery cost, but here we take both effects into account.

However, we stress that the suitability of the solution approach for practical, industry-sized application hinges on the ability to identify feasible slots F_{ai} and to solve the online pricing problem very quickly, namely in less than 100 milliseconds, as advised by our industry partner. Furthermore, close to the delivery day, we may have orders arriving in quick succession, so there may be no time for offline computations between order arrivals.

In the following sections, we propose a solution approach that adheres to these tight practical limitations. The delivery cost function approximation is discussed in Section 4, and the value function approximation in Section 5.

4. Delivery cost approximation

Evaluation of the delivery cost function $C(\vec{x}, \vec{y})$ requires a solution to the capacitated vehicle routing problem with time windows, which is known to be NP-hard. Moreover, we need to evaluate this cost function repeatedly for various states (\vec{x}, \vec{y}) . However, we only need a reasonable cost estimate; the underpinning routes themselves are not required for our solution approach. Therefore, we propose using the clustering-first, route-second strategy developed by Daganzo (1987), which considerably simplifies the problem whilst still retaining

sufficient information to provide a useful cost estimation, as we demonstrate numerically in Section 7. This allows us to solve the dynamic program (3.1) approximately. We emphasize that this cost estimation is different from the final cost assessment used in our simulation experiments given a final set of orders for which we do base the costs on specific routes.

The idea is to decompose the problem geographically by assuming that each area has a single delivery van associated with it, and that this van is driving to complete a full cycle in each time slot within its designated area. Under some further assumptions, we can express the daily traveling distance D_a within a given area a by a simple function of the number of orders received, \vec{x}_a . These assumptions are that customer locations are randomly and nearly uniformly scattered within the area, that the time windows are equally long, that customers only place requests within one of the time periods, and that demand is balanced over time slots. The assumptions regarding uniformly-distributed customers over space and time are strong. Whilst we would expect that our dynamic pricing approach will eventually lead to more uniformly-distributed orders over time, some peaks and troughs are still likely to occur. Likewise the geographical distribution will typically be non-uniform.

Therefore, one would expect the resulting estimates to be of limited quality, in so far as the actual vehicle routes would be much more flexible. However, our main objective is to devise an approach that requires very little computational time to evaluate routing costs, yet that estimates them in such a way as to still improve overall profitability. We define a single set of delivery areas for the delivery day under consideration, and we keep this fixed over the entire booking horizon. Whilst being restrictive, this approach allows us to (approximately) evaluate delivery costs very quickly.

Clustering: defining delivery areas

First, we define rectangular delivery areas a with length L_a and width W_a that represent geographical clusters of customers to be served. The clustering problem features routing constraints; hence, approaches like k-means cannot be applied in a straightforward manner. Any area must be defined such that a single van can accommodate all expected orders (capacity constraint), and such that it is small enough to allow the van to complete full cycles and to visit and serve all customers in each delivery time slot (time constraints).

We propose the following clustering approach. For a given weekday, e.g. Monday, we obtain the average number of daily deliveries N_z in postcode z from the final delivery schedules

of past Mondays. For any postcode center located in a given rectangular area, this area's van needs to serve all daily orders associated with these postcodes. The total daily number of orders in area a is denoted by $N_a = \sum_{z \in a} N_z$, where $z \in a$ denotes all postcodes centered within area a . These N_a orders form our expectation of total daily demand for this area; according to our assumptions above, we expect $\lceil N_a/m \rceil$ orders in each of the m time slots.

The overall delivery region is partitioned into bands of equal size. A delivery area is defined as a piece of a band that satisfies both delivery time and capacity constraints. We only need to decide on the width of each area because its length is fixed by the latitude of each band. For illustration, Figure 1 (in Section 7) shows such a cluster derived for the Greater London region. In that numerical study, we tested different numbers of these horizontal bands and selected the one resulting in the smallest number of required vans.

For a given set of bands, we move from east to west within a given band to determine the maximum allowable width W_a of a delivery area a that satisfies:

- the time constraint that the total traveling and service time required to serve all $\lceil N_a/m \rceil$ orders in a slot should not exceed the duration of that slot (e.g. one hour). We assume a known average traveling speed v of the delivery van and a known average service time τ at a customer location. Distance driven is split into vertical and horizontal distances: in each time slot s , the van travels the whole length L_a of the area twice. Note that the expected distance of two uniformly-distributed points in the unit interval is $1/3$. The van is assumed to travel in a full cycle covering the upper half of the rectangle on the first half cycle, and the other on its return, so the expected distance between two orders is $W_a/6$. This results in the constraint:

$$(2L_a + \frac{W_a}{6} \lceil N_a/m \rceil)/v + \tau \lceil \frac{N_a}{m} \rceil \leq 1.$$

We refer the interested reader to Daganzo (1987) for further details of the derivation of this formula.

- the capacity constraint, namely that the total number of totes does not exceed the capacity κ of the delivery van: $N_a \sum_{i \in I_a} i \sigma_{ai} \leq \kappa$.

This method produced realistic numbers of required vans in our numerical experiments.

Routing: approximating delivery costs

For a rectangular area a of width W_a and length L_a , we define the stem distance ρ_a from the depot to the area's center. For a given number of $\sum_s x_{as}$ orders to be served on the delivery day under consideration, the daily traveling distance is given by

$$D_a(\vec{x}_a) = (2\rho_a - L_a) + \sum_{s=1}^m (2L_a\delta_{as} + x_{as}W_a/6), \quad (4.1)$$

where

$$\delta_{as} = \begin{cases} 0, & \text{if } x_{as} = 0, \\ 1, & \text{if } 0 < x_{as} \leq M_a, \\ \infty, & \text{if } x_{as} > M_a, \end{cases}$$

and the maximum number of orders that can be served within a one hour time slot is denoted by $M_a := \operatorname{argmax}\{x \in \mathbb{Z}^+ | \tau x + \left[2L_a + \frac{W_a}{6}x\right] \frac{1}{v} \leq 1\}$. The first part of the formula expresses the stem distance traveled between the depot and the area, and the second part is based on the same reasoning as presented for the time constraint in the clustering discussion above.

We assume that driving distance is the only cost incurred in accepting deliveries since the fleet and the drivers' salaries are assumed to be fixed costs, so the total cost of area a is $C_a(\vec{x}_a, y_a) = \xi D_a(\vec{x}_a)$ if the van capacity is not exceeded, i.e. $y_a \leq \kappa$ (and ξ is a known cost-per-mile factor), or $C_a(\vec{x}_a, y_a) = \infty$ otherwise.

In summary, this delivery cost estimation has the advantage that the resulting overall cost function $C(\vec{x}, \vec{y}) = \sum_a C_a(\vec{x}_a, y_a)$ is decomposable by delivery area, and can be quickly evaluated. We exploit these features in our approximation of the dynamic programming value function.

5. Value function approximation

For the final stage $T+1$, the value function decomposes by areas: $V_{T+1}(\vec{x}, \vec{y}) = -\sum_a C_a(\vec{x}_a, y_a)$. Since slot-pricing decisions are independent between different areas, the dynamic program as a whole decomposes by area:

$$\begin{aligned} V_t^a(\vec{x}, \vec{y}) = \max_{\vec{d}_a} \lambda \sum_i \mu_a \sigma_{ai} \sum_{s \in F_{ai}} P_{s, F_{ai}}(\vec{d}_a) & \left[ir + d_{as} - \left(V_{t+1}^a(\vec{x}_a, y_a) - V_{t+1}^a(\vec{x}_a + \mathbf{1}_s, y_a + i) \right) \right] \\ & + V_{t+1}^a(\vec{x}, y_a) \quad \forall (\vec{x}_a, y_a) \in \mathcal{X}_a, \end{aligned} \quad (5.1)$$

where $\mathcal{X}_a = \{(\vec{x}_a, y_a) \in (\mathbb{N}^m \times \mathbb{N}) | 0 \leq x_{as} \leq M_a, y_a \leq \kappa\}$. In the following, we omit area index a since we are focusing on a solution to this single-area dynamic program. It is still intractable because of the large state space which grows exponentially in the number of time slots; therefore, we propose to use approximate dynamic programming with a linear value function approximation, similar to the affine approximation proposed by Adelman (2007). Note that we omit dependence on the total number of totes y to be delivered in the area under consideration. This number is relevant to the cost function in determining whether all orders fit into the van, and is used in the determination of slot feasibility. However, the time constraints are usually much more restrictive than van capacity if the delivery time slots are narrow (say, one hour), so we ignore y in the approximation:

$$V_t(\vec{x}, y) \approx \bar{V}_t(\vec{x}) := \gamma_0 - \sum_s \gamma_s x_s + (T + 1 - t)\theta, \quad \forall (\vec{x}, y) \in \mathcal{X}. \quad (5.2)$$

Parameter γ_s can be interpreted as an estimate of the opportunity cost of accepting an order in slot s (regardless of order size i) since $V_{t+1}(\vec{x}, y) - V_{t+1}(\vec{x} + \mathbf{1}_s, y + i) \approx \gamma_s$. Parameter θ reflects the time dependence of the value function: the fewer time periods remain until the end of the booking horizon, the smaller the expected profits that can be gained over these remaining time periods.

In Algorithm 1, we outline our proposed approximate dynamic programming procedure to find parameters $\vec{\gamma}$ and θ . We sample k^{\max} paths of order arrivals indexed by k , and use our current best knowledge of the parameters to approximate the value-to-go in the dynamic programming recursion. This allows us to step forward in time, and at each time step t we calculate the value $\hat{V}_t^{(k)}$ of being in state $\vec{x}_t^{(k)}$. Next, we update the parameters with a stochastic gradient step. Specifically, we seek to find parameters that bring our value function approximation $\bar{V}_t^{(k)}$ closer to the observed value $\hat{V}_t^{(k)}$:

$$\min_{\vec{\gamma}, \theta} \frac{1}{2} \mathbb{E} \left[\bar{V}_t^{(k)}(\vec{x}_t^{(k)}) - \hat{V}_t^{(k)} \right]^2.$$

The updating scheme (with fixed step sizes $\alpha_1, \alpha_2, \alpha_3$ along the negative gradient directions) is thus the following:

$$\begin{aligned} \gamma_0^{(k)} &= \gamma_0^{(k-1)} - \alpha_1 [\bar{V}_t^{(k-1)}(\vec{x}_t^{(k)}) - \hat{V}_t^{(k)}], \\ \gamma_s^{(k)} &= \gamma_s^{(k-1)} - \alpha_2 [\bar{V}_t^{(k-1)}(\vec{x}_t^{(k)}) - \hat{V}_t^{(k)}] x_{ts}^{(k)}, \quad \forall s, \\ \theta^{(k)} &= \theta^{(k-1)} - \alpha_3 [\bar{V}_t^{(k-1)}(\vec{x}_t^{(k)}) - \hat{V}_t^{(k)}] (T + 1 - t). \end{aligned}$$

We stop the procedure after k^{\max} iterations (corresponding to the k^{\max} sample paths), and use the final value function approximation in the real-time control policy to make pricing decisions, as discussed in the next section.

6. Real-time control policy

As soon as a customer request arrives, we need to determine which delivery slots are feasible, and then decide on the delivery price. In fact, for the sake of practical relevance, this decision needs to be made almost instantaneously. We propose to determine the area clusters and the corresponding value function approximations before the start of the booking horizon.

Within the booking horizon, we can then check the “feasibility” of delivery in a given area a by checking whether (a) the current number of totes to be delivered in this area exceeds the van’s capacity, and (b) whether we exceed the maximum number of orders M_a in any slot. Both conditions are simple comparisons of known numbers.

This proposed “feasibility” check is unrealistic, in so far as the actual routing would look very different from the assumed area-based routing approximation. However, it may be regarded as a conservative estimate: whilst slots deemed “infeasible” may actually be feasible in the actual vehicle routing process, “feasible” slots would be expected to indeed be feasible. Keeping sets of feasible vehicle routings, as done by Yang et al. (2016), is likely to be too time-consuming for real-time decision making.

Next, given the set of feasible slots $F_{ai}(\vec{x}, \vec{y})$ for the incoming request in area a for i totes and associated order profit ir given state (\vec{x}, \vec{y}) , we need to find the optimal delivery charges to offer. We are seeking to solve the following problem:

$$\vec{d} = \operatorname{argmax}_{s \in F_{ai}} \sum P_{s, F_{ai}}(\vec{d}_a) \left[ir + d_{as} - \gamma_s \right]. \quad (6.1)$$

The difficulty of solving this problem depends on the choice model underpinning the customers’ time slot decisions as well as on the range of feasible price vectors. Note that essentially the same problem needs to be solved repeatedly in the approximate dynamic programming iterations as described in Algorithm 1. Various constellations have been investigated and tractable formulations proposed, for example using the MNL choice model and continuous prices (Dong et al., 2009), MNL with a discrete price set (Davis et al., 2013), or nested MNL with bounded continuous prices (Rayfield et al., 2013).

We use an MNL choice model with continuous prices, as in Yang et al. (2016), whereby the probability of a customer choosing delivery slot s , given that the set F of slots is available at prices \vec{d} , is defined by:

$$P_{s,F}(\vec{d}) = \frac{\exp(\beta_0 + \beta_s + \beta_d d_s)}{\sum_{k \in F} \exp(\beta_0 + \beta_k + \beta_d d_k) + 1},$$

where β_0 is an offset parameter, β_s measures the attractiveness of slot s and β_d is the price sensitivity. The no-purchase utility is normalized to zero.

Proposition 6.1. *For the MNL choice model and continuous prices, the optimal solution to (6.1) is given by*

$$d_s^* = \gamma_s - ir - \frac{h}{\beta_d}, \quad \forall s \in F,$$

where h is the unique solution of

$$(h - 1) \exp(h) = \sum_{s \in F} \exp(\beta_0 + \beta_s + \beta_d(\gamma_s - ir)). \quad (6.2)$$

Proof. Theorem 1 in Dong et al. (2009). □

Standard Newton root search can be employed to find h in (6.2).

7. Numerical results

We tested our approach in a simulation study based on real data from our industrial partner. With these experiments, we sought answers to the following questions:

1. Does the proposed policy deliver consistent improvement in profitability over various demand scenarios compared with benchmarking policies? How does dynamic pricing with different opportunity cost estimates perform, and what is the value of including order displacement cost in the opportunity cost estimate?
2. Is computational speed sufficient for potential commercial application?
3. Can we glean managerial insights from the new pricing approach regarding whether delivery areas that are far from the depot should be priced differently from closer areas, and how order volume influences the average delivery charge?

We discuss these questions and summarize our findings in the next sub-sections.

7.1. Data description

The data were provided by a major e-grocer in the United Kingdom, focussing specifically on delivery operations in the Greater London area. The same data were used by Yang et al. (2016) to estimate time-dependent customer arrival processes and to calibrate an MNL choice model. The dataset contains anonymized customer booking requests over six months from the beginning of June to the end of November 2011, all made through the company’s website. Customer are identified by a unique ID number, and they must be logged into their personal account in order to book delivery. The postcode of each customer is contained in the data. Every request to display available delivery slots is stored, including which customer made the request, the order size in terms of number of totes, the time and delivery day of the request, which slots were displayed as available, and at what charges. Customers’ delivery slot decisions are likewise recorded. Expenditure is not included in the data set, but a fixed average revenue per tote of £30.39 was provided. Furthermore, since we aimed to maximize profit, we assumed the average profit before delivery costs to be 30 per cent of revenue.

Regarding the range of data used, only customers who had to pay per delivery were considered, excluding those with a subscription for free delivery. Furthermore, we focussed our attention on customers wanting to book a Monday delivery. Customers who considered a Monday delivery but then decided on delivery on another day were considered to be lost sales because we optimized for each day individually. It would have been desirable to include multiple days in the choice model, but this would have considerably increased its complexity and is beyond the scope of this paper. Twenty-six booking histories were associated with Monday deliveries over the full time horizon available, each containing several thousand customer arrivals. We only used the latest instance of a booking request by a particular customer for a particular delivery day (sometimes a customer looked at options for the same delivery day at different times). For the sake of simplicity, we did not include cancelations in the model; thus, we removed canceled orders from the data.

The arrival process was estimated over $T = 6,990$ non-homogeneous periods with arrival probability $\lambda = 0.824$ in each period. Note that time period sizes were chosen to allow for constant probability: early periods were wide, whereas periods close to the delivery day were narrow (see Yang et al. (2016) for details of the estimation procedure). Since the data contained the postcode of each customer, we were able to calculate the probability μ_z that

a new delivery request would hail from a particular postcode z as the proportion of requests from z relative to all requests. This probability was then used to derive the probability μ_a that an arriving customer would belong to area a (consisting of a collection of postcodes).

Regarding delivery slot choices, the online grocer uses 27 partly overlapping delivery time slots of one-hour duration, starting either on the hour or at half past the hour. However, our routing cost estimation assumes that we have non-overlapping slots. Therefore, we transformed the data by randomly changing any half-past-the-hour slot request to either the preceding or succeeding overlapping slot with equal probability. We estimated the MNL choice model based on these modified data and obtained parameters as reported in Table 1. For details of how to derive the MNL parameters and the arrival process, see Yang et al. (2016). Note that the base utility parameter is small relative to the utility of the non-purchase option (which is set to zero). This is because the data contains many cases where customers looked at a particular day but then went on either to select a slot on a different day or not to book delivery at all. The β parameters reflect the popularity of different slots; for example, 9-11am, noon-1pm, and 9-10pm are particularly attractive. The price sensitivity parameter β_d is negative in line with expectation. All delivery charges contained in the data belong to the set $\{\pounds 0, \pounds 1, \dots \pounds 7\}$.

β_6	β_7	β_8	β_9	β_{10}	β_{11}	β_{12}	β_{13}	β_{14}
-1.0305	-0.3591	0.3107	0.5922	0.6154	0.0796	0.5356	-0.2415	-0.6286
β_{15}	β_{16}	β_{17}	β_{18}	β_{19}	β_{20}	β_{21}	β_{22}	
-1.6736	-0.4351	-0.161	0	0.2533	0.0736	0.562	0.2346	

Table 1: MNL parameters estimated for Monday deliveries. Base utility $\beta_0 = -2.5087$, price sensitivity parameter $\beta_d = -0.0766$. Slot preference β_6 represents the 6-7am slot, etc. Slot 18 is used as a reference point and is hence set to zero.

Finally, as inputs for the delivery cost estimation we used $\xi = 0.25$ as the cost per mile, and $\kappa = 80$ totes as the van capacity. Stem distances ρ_a from the depot to the center of a given rectangular area a were calculated as the crow flies. Average van speed was assumed to be $v = 25.4$ miles per hour as provided by our industry partner. Deliveries are made in two sequential shifts (6:00 to 15:00, 15:00 to 23:00). We made the simplifying assumption that vans were ready for delivery at the beginning of each shift; however, we did include the driving cost

between each area and the depot in the cost estimation because we were mainly interested in obtaining reasonable delivery cost estimations and were less concerned with feasible schedules. Our area-based cost estimation is anyway a very conservative approximation, so feasibility under this scheme is likely to imply feasibility under a more sophisticated routing.

7.2. Clustering

We derived the delivery area definitions as described in Section 4. The overall delivery region was divided into 16 bands, resulting from experimenting with different numbers to see which resulted in the least number of areas required to cover all postcodes z with positive average total daily number of orders, denoted by N_z . For our data, using 16 bands and applying our ad hoc area definition method resulted in 111 areas. The resulting rectangular areas and the customer locations covered by each are shown in Figure 1. Some areas on the outskirts of London are wide and contain few customer locations; others in central London are very small due to the high density of customers there. Recall that the maximum width of an area is determined using the time and capacity constraints described in Section 4; therefore, in areas with high order density, the time constraint becomes binding even for small values of width W_a . Each dot represents a postcode z with a certain underpinning average daily number of orders N_z . We assume that these orders are all uniformly distributed over the m delivery slots (required for the cost estimation framework of Daganzo (1987)), so here we do not use historical slot booking information. Note that some areas span white space because we need to be able to attribute an order from a postcode that is in the delivery region but from which we have not yet received orders to a specific area. The area definition remains static throughout the booking horizon, and therefore needs to be able to accommodate orders from any part of the delivery region.

All experiments were conducted on the basis of this area decomposition. The probability that an arriving customer will be from area a is defined by $\mu_a := \sum_{z \in a} \mu_z$, where $z \in a$ is shorthand notation to represent all postcode centers within area a . For each postcode z , we know the average order size i_z (measured by the number of totes and rounded to the nearest integer) for the delivery day under consideration. The order size distribution σ is assumed to be the same for each cluster, and is defined by $\sigma_i := \sum_{z: i_z = i} \mu_z$, for all $i \in \{1, 2, \dots, 10\}$.

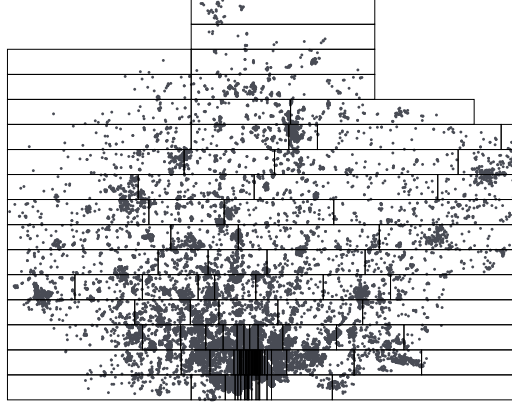


Figure 1: Scatter plot of customer location clusters. Each dot represents a zip code where at least one order has been received in at least one of the 26 booking histories.

7.3. Simulation results

Using arrival rates λ , μ_a and order size distributions σ , we generate 1,000 arrival streams over the entire booking horizon T . Order value is derived from the sampled order size multiplied by the average revenue per tote. For these 1,000 demand scenarios, we test the following policies:

- *VS*: Value-based, Static pricing. Delivery charge is based on order value, namely £3 for goods worth £50 or more, and £5 otherwise.
- *F4, F5*: Fixed prices, at £4 and £5 respectively, for all time slots.
- *OC-0*: Given a request for delivery of an order of size i to area a (and a feasible set of slots F_{ai}), assume that the **opportunity cost** is **zero** and solve (6.1), with γ_s replaced by 0 for all slots s .
- *OC-C*: Given a request for delivery of an order of size i to area a (and a feasible set of slots F_{ai}), assume that the **opportunity cost** equals the marginal estimated delivery **cost** and solve (6.1) with γ_s replaced by $\xi(2L_a + W_a/6)$ for the first order in slot s , and by $\xi W_a/6$ subsequently.
- *OC-R*: Given a request for delivery of an order of size i to area a (and a feasible set of slots F_{ai}), assume that the **opportunity cost** equals γ_s , where the boundary condition in Algorithm 1 is replaced with $\bar{V}_{T+1}^{(k)}(\vec{x}) = 0$ for all \vec{x} , for all k (i.e. we ignore all cost

implications). The underpinning approximate dynamic program uses step sizes α of 0.0001, 0.00014 and 0.00025 for γ_0 , θ and γ_s , respectively (note that different scaling of these variables requires different step sizes). We use $k^{\max} = 3,000$ iterations.

- *OC-CR*: Given a request for delivery of an order of size i to area a (and a feasible set of slots F_{ai}), assume that the **opportunity cost** equals γ_s , i.e. it consists of both **cost** and **revenue** effects, and solve (6.1). This is our proposed approach. All parameters of Algorithm 1 are the same as for *OC-R*.

For all policies, feasibility of delivery in a particular area, for a particular slot at a particular time t in the booking horizon is evaluated based on the fixed area definition, as described in Section 6. Customers' choices are sampled from the MNL model depending on the delivery charges that we limit to the interval $[-£10, £10]$ as in Yang et al. (2016). We follow their approach of projecting optimal prices onto this interval, so the resulting price set will no longer necessarily be optimal.

We remark that *OC- θ* is a myopic policy which assumes that there are no delivery capacity constraints and no delivery costs; accordingly, this policy should do well in areas where expected demand is much lower than capacity and where delivery costs are low. Policy *OC-C* should do better, in that it accounts for delivery costs, and overall is also expected to work well if expected demand is much less than capacity.

Final delivery costs are calculated in a two-stage process: we first assume that each vehicle serves one area only and insert as many accepted orders as possible into the single-cluster delivery route using a greedy insertion heuristic. If any orders remain unserved, we then try in a second stage to insert these orders into the delivery routes of vans in adjacent areas that still have available capacity. If there are still orders left unserved, then we add a fixed penalty cost of £5 to each (corresponding to standard second-class delivery by the mail service).

In Table 2, we report the results over all delivery areas for different scalings of the arrival rate λ , so as to gain insights into the robustness of the profit improvements of our proposed policy *OC-CR* over the benchmark *VS* with respect to changes in demand intensity. For each scaling scenario, the approximate dynamic program is only solved once and the resulting parameters $\vec{\gamma}$ are used in the *OC-CR* policy in all simulations.

Table 2 provides many insights. First and most importantly, we observe that *OC-CR* performs consistently the best over all scenarios, with improvements over the benchmark in

the range of 2.2–2.5 per cent. Secondly, we always observe $OC-\theta < OC-C < OC-R < OC-CR$.

In particular, $OC-CR$ is significantly and consistently better than $OC-C$ in all scenarios, demonstrating the value of incorporating the impact of future profit opportunities from orders, rather than just the estimated marginal delivery cost, into the opportunity cost. $OC-CR$ achieves these profit improvements despite collecting fewer orders than the other two dynamic policies, and at a higher cost per order. This is due to our focus on optimizing total profit rather than minimizing costs. The algorithm anticipates future order values and in which time slot they are likely to occur, so that the pricing can influence demand accordingly.

The $OC-R$ policy represents a tractable implementation of the dynamic programming approach of Asdemir et al. (2009). The observed results are intuitive. First, $OC-R$ works better than all other simple policies apart from $OC-CR$ as long as demand is high relative to capacity (so the expected value of displaced order revenue is often positive). $OC-R$ gives the highest average price within the dynamic pricing group, as a result of the higher expected opportunity cost without considering delivery costs. As the demand scaling factor increases, $OC-R$ and $OC-CR$ become increasingly similar. This is due to the increasing importance of order value over delivery cost: when demand is high, by accepting an order in a certain slot we are more likely to displace future orders than in a scenario with low demand. Therefore, the proportion of expected displaced future profits becomes bigger relative to routing costs as demand increases (assuming a fixed fleet), and so $OC-R$ behaves almost like $OC-CR$ under high-demand scenarios.

All dynamic policies attract higher average order values than static policies, but overall profitability of $OC-\theta$ and $OC-C$ may suffer compared with the benchmark, even if they attract more orders. This is because they set the delivery charge too low (due to underestimating the opportunity cost), and hence the additional profit from orders fails to compensate for resulting loss in delivery charge income, resulting in an overall loss. This demonstrates the importance of the revenue stream from delivery pricing (in the absence of other potential forms of control such as slot availability control). Dynamic pricing per se does not improve profitability over simple fixed-price policies; its success hinges on good opportunity cost estimates that capture both marginal delivery costs and future order displacement costs.

The lower the demand relative to available capacity (i.e. the smaller the scaling parameter), the better $OC-\theta$ and $OC-C$ perform because the true opportunity costs move toward the

marginal delivery cost (note that with demand considerably below capacity, it is unlikely that future orders will be displaced by an order). Accordingly, we expect that if demand is scaled down further, then at some point *OC-C* and *OC-CR* will perform similarly well.

We emphasize that the observation that *OC-C* may be worse than *VS* does not contradict the findings of Yang et al. (2016), who propose a policy that approximates opportunity costs only with estimated marginal delivery cost and who observe that this policy improves over the same fixed-price policy. Their opportunity-cost estimation relies on a computationally more intensive way of estimating marginal delivery costs. This is likely to produce better results than *OC-C* and static pricing policies, but it would be difficult to implement in a real-time decision-making environment. We limit our comparison to policies that we deem suitable for real-time decision making in large-scale applications.

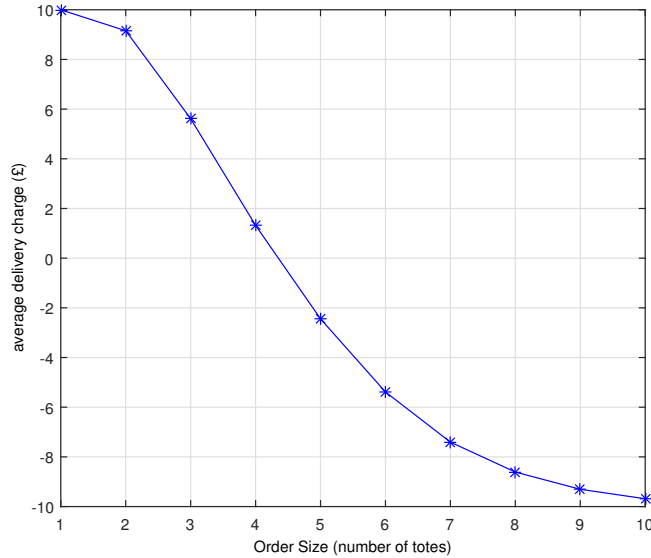


Figure 2: Average delivery charge of *OC-CR* for different order sizes.

Figure 2 depicts how delivery charges vary with order size when using the *OC-CR* policy. The average delivery charge of *OC-CR* is calculated over all requests with the same order size (expressed in the number of totes) for the scenario with a scaling parameter of 1. As one would expect, the larger the order, the lower the charge tends to be; in fact, for larger orders, we often even charge negative delivery fees, i.e. discounts. The graph reflects that the algorithm differentiates the value of orders and adjusts pricing accordingly.

The opportunity cost estimates γ_{as} of *OC-CR* may be interpreted as the average profit

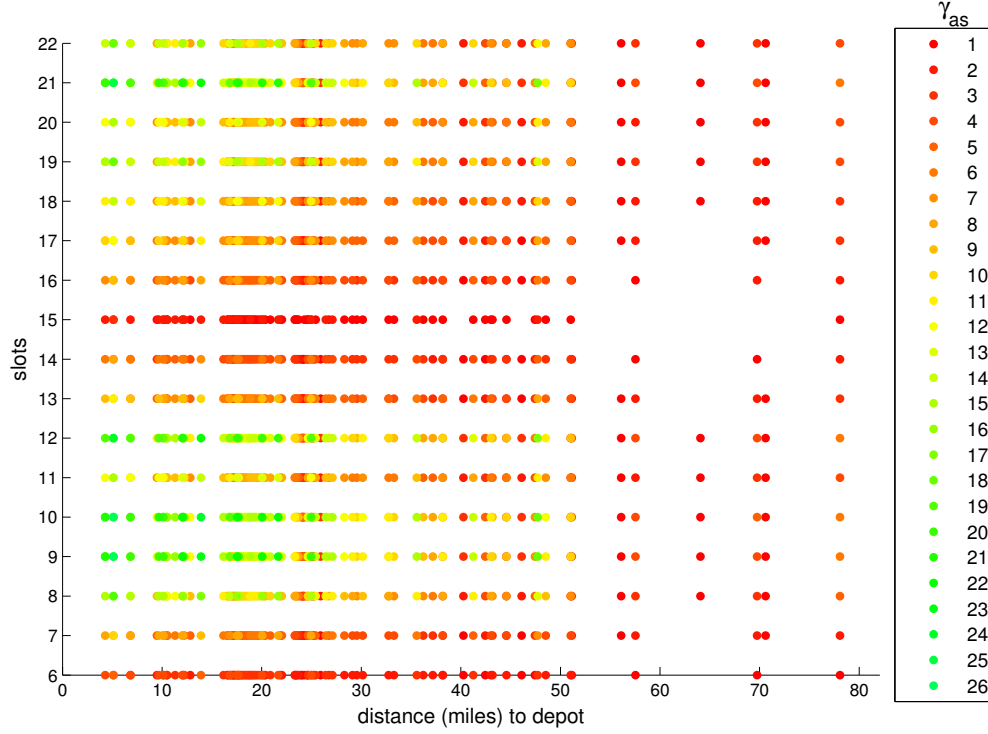


Figure 3: Opportunity cost estimate γ_{as} of *OC-CR* policy for each area a and time slot s . Missing dots represent $\gamma_{as} = 0$. The color represents the value of γ_{as} : high opportunity cost values are shown in green, low values in red.

value that slot s has in a ; in other words, we do not want to have a customer book that slot unless we are making at least γ_{as} profit from that order and the delivery charge combined. The estimates take the popularity of time slots into consideration, as well as the likelihood that a future order will be displaced by accepting an order into a time slot. Figure 3 shows that popular slots (9am-10am, noon-1pm, 7-10pm) receive high opportunity cost estimates in areas close to the depot. The depot is located in London, so delivery areas close by (i.e. within around 30 miles) have a denser customer population than more remote ones; hence, demand for peak-time delivery slots is likely to reach full capacity.

Figure 3 also illustrates that remote areas have very low opportunity cost estimates across all slots. This may be somewhat counter-intuitive since they are also associated with high stem driving costs, which are taken into account by the *OC-CR*. However, demand in these areas is much smaller than available van capacity; therefore, the opportunity cost is influenced mainly by delivery costs and less by order displacement costs. The latter often have the biggest

impact on γ_s , and *OC-CR* accordingly produces low estimates for remote areas. Delivery charges will also tend to be lower than in busy areas so as to attract more orders, a feature of our approach that may help to develop rural markets. Note that delivery charges are one of the main inhibitors of online shopping: “26% of consumers who have either stopped or are shopping less for groceries online said they are doing so because of higher delivery charges”, see Mintel (2016).

The fact that the opportunity cost estimates γ_{as} are close to zero in remote areas also means that *OC-0* should give similar results to *OC-CR* for these areas. Therefore, one might consider using the simpler policy *OC-0* in these areas so as to further reduce computational effort.

In terms of computational effort, parameter estimation through approximate dynamic programming for all area clusters takes a total of around 15 to 17 minutes (depending on the scaling of the arrival rate) on a standard desktop PC. More important is the speed of online feasibility and pricing decisions that must be made when a customer booking request arrives. The critical threshold for practical implementation is stated by our industrial partner to be 100 milliseconds; our method achieves it in an average of 0.4 microseconds.

In summary, our findings regarding the research questions listed at the outset of this section are as follows:

1. Consistent improvement of *OC-CR*: We observe profit improvements of over two per cent across all demand scenarios over the *VS* benchmark policy. This suggests that the method returns stable improvements relative to uncertain demand intensity. Dynamic pricing policies *OC-O* and *OC-C* may perform worse than static pricing because they may set prices too low if the opportunity cost is over-influenced by order displacement cost (especially when capacity is tight).
2. Computational speed: The crucial online calculations underpinning the feasibility check and pricing decisions are virtually instantaneous owing to the simplistic routing model and efficient price optimization.
3. Managerial insights: Although one might intuitively expect that delivery charges in areas distant from the depot should be higher than in areas nearby, our approach is designed such that the best pricing policy is to keep charges low in remote areas so as to increase overall demand there. This is also beneficial with respect to perceived fairness;

customers would not accept being penalized for living further from the company’s depot. Charges should reflect the value of a delivery time slot, in terms of both marginal delivery costs and future order displacement costs. Dynamic pricing may perform much worse than static pricing if the opportunity cost does not include order displacement costs, especially when demand is similar to or exceeds capacity.

If profit is the overall objective, performance should not be measured in terms of cost per order, number of deliveries made or average order profit (before delivery). Delivery charges may make a substantial contribution to overall profit.

8. Conclusions and future research

We propose a new method of controlling demand through delivery time slot pricing in attended home delivery management with a focus on developing an approach suitable for industry-scale implementation. To that end, we exploit a relatively simple yet effective way of approximating delivery costs by decomposing the overall delivery problem into a collection of smaller, area-specific problems. This cost estimation serves as an input to an approximate dynamic programming method, which provides estimates of the opportunity cost of accepting a given customer booking in a specific time slot. These estimates depend on the area and on the delivery time slot under consideration.

Using real, large-scale industry data, we estimate a demand model involving a multinomial logit choice model, and show in simulation studies that we can improve profits by over two per cent in all instances compared with using an order-value-dependent, fixed-price policy. These improvements are achieved despite making strong assumptions in delivery-cost estimation, which are needed to reduce computational run-time to a level that allows real-time decision making. Our approach provides quantitative insight into the importance of incorporating expected future order displacement costs into opportunity-cost estimation alongside marginal delivery costs.

There are some limitations of our study that warrant further research. First, it would be insightful to conduct a simulation study in which the final delivery cost incurred at the end of each simulated booking horizon is based on an industry-standard vehicle-routing solution. Second, our proposed approach uses opportunity-cost estimates that depend neither on time of booking within the booking horizon nor on the level of orders accepted; Meissner and

Strauss (2012) show that incorporating this intuitive dependence may further improve the results at the cost of significantly increased computational burden. Our model may take time-dependence to some extent into account by re-solving several times over the booking horizon with the most recent information on accepted orders; we thus obtain opportunity-cost estimates that may change over time. Third, one would expect a better value function approximation to return stronger results, such as a piecewise linear approximation. Finally, it would be desirable to include a choice between adjacent delivery days in addition to a choice between delivery time slots. As we remarked in Section 6, a nested logit model might be used to that end, as long as the resulting pricing problem could still be solved sufficiently quickly.

References

- Adelman, D., 2007. Dynamic bid prices in revenue management. *Operations Research* 55 (4), 647–661.
- Agatz, N., Campbell, A., Fleischmann, M., Savelsbergh, M., 2011. Time slot management in attended home delivery. *Transportation Science* 45 (3), 435–449.
- Agatz, N. A. H., Fleischmann, M., van Nunen, J. A. E. E., 2008. E-fulfillment and multi-channel distribution: A review. *European Journal of Operational Research* 187 (2), 339–356.
- Asdemir, K., Jacob, V. S., Krishnan, R., 2009. Dynamic pricing of multiple home delivery options. *European Journal of Operational Research* 196 (1), 246–257.
- Bell, D., Song, S., 2007. Neighborhood effects and trial on the internet: Evidence from online grocery retailing. *Quantitative Marketing and Economics* 5 (4), 361–400.
- Bühler, D., Klein, R., Neugebauer, M., 2016. Model-based delivery cost approximation in attended home services. *Computers & Industrial Engineering* 98, 78–90.
- Campbell, A. M., Savelsbergh, M., 2006. Incentive schemes for attended home delivery services. *Transportation Science* 40 (3), 327–341.
- Campbell, A. M., Savelsbergh, M. W. P., 2005. Decision support for consumer direct grocery initiatives. *Transportation Science* 39 (3), 313–327.

- Cleophas, C., Ehmke, J. F., 2014. When are deliveries profitable? Considering order value and transport capacity in demand fulfillment for last-mile deliveries in metropolitan areas. *Business and Information Systems Engineering* 6 (3), 153–163.
- Daganzo, C. F., 1987. Modelling distribution problems with time windows: Part 1. *Transportation Science* 21(3), 171–179.
- Davis, J., Gallego, G., Topaloglu, H., April 2013. Assortment planning under the multinomial logit model with totally unimodular constraint structures, working paper, Cornell University, Ithaca, NY.
- Dong, L., Kouvelis, P., Tian, Z., 2009. Dynamic pricing and inventory control of substitute products. *Manufacturing & Service Operations Management* 11 (2), 317–339.
- Ehmke, J. F., Campbell, A. M., 2014. Customer acceptance mechanisms for home deliveries in metropolitan areas. *European Journal of Operational Research* 233 (1), 193–207.
- Hübner, A., Kuhn, H., Wollenburg, J., 2016. Last mile fulfillment and distribution in omnichannel grocery retailing. *International Journal of Retail & Distribution Management* 44 (3), 228–247.
- Klein, R., Mackert, J., Neugebauer, M., Steinhardt, C., 2016. On the approximation of opportunity cost for dynamic pricing in attended home delivery, working paper, University of Augsburg, Germany.
- Klein, R., Neugebauer, M., Ratkovitch, D., Steinhardt, C., 2015. Differentiated time slot pricing under routing considerations in attended home delivery, working paper, University of Augsburg, Germany.
- Lang, M., Ehmke, J., Köhler, C., Cleophas, C., 2017. Considering equity in retail order acceptance for attended deliveries. In: *Annual Conference of the Pricing & Revenue Management Section, Gesellschaft für Operations Research, Zurich, Switzerland*.
- Meissner, J., Strauss, A., 2012. Network revenue management with inventory-sensitive bid prices and customer choice. *European Journal of Operational Research* 216 (2), 459–468.
- Mintel, 2016. Online Grocery Retailing UK – March 2016. Tech. rep., Mintel Group Ltd, London.

- Rayfield, W. Z., Rusmevichientong, P., Topaloglu, H., September 2013. Approximation methods for pricing problems under the nested logit model with price bounds, working paper, Cornell University, Ithaca, NY.
- Yang, X., Strauss, A. K., Currie, C. S. M., Eglese, R., 2016. Choice-based demand management and vehicle routing in e-fulfillment. *Transportation Science* 50 (2), 473–488.

Algorithm 1 Approximate Dynamic Programming Procedure.

- 1: Initial value function parameters: $\vec{\gamma} \leftarrow \vec{0}$, $\theta \leftarrow 0$ to define $\bar{V}_t^{(0)}(\vec{x})$ in (5.2) for all t , \vec{x} .
 - 2: Boundary condition: $\bar{V}_{T+1}^{(k)}(\vec{x}) = -\xi D(\vec{x})$ for all \vec{x} , $k = 1, \dots, k^{\max}$, where ξ is cost per mile and $D(\vec{x})$ is the total milage driven
 - 3: Iteration counter: $k \leftarrow 1$
 - 4: Initial state: $\vec{x}_1^{(k)} = \vec{0}$, $y_1^{(k)} = 0$ (initially no orders on record)
 - 5: **while** $k \leq k^{\max}$ **do**
 - 6: Generate sample path of order arrivals (for area under consideration): $(\vec{R}_1^{(k)}, \dots, \vec{R}_T^{(k)})$, where $\vec{R}_t^{(k)}$ is a vector containing either zeros (no order), or information on order size $i^{(k)}$, and its profit $r_t^{(k)}$ before delivery cost.
 - 7: **for all** $t = 1, 2, \dots, T$ **do**
 - 8: Define feasible set $F_i := \{s | \vec{x}_{ts}^{(k)} + \mathbf{1}_s \leq M_a\}$ if there is sufficient van capacity ($y^{(k)} + i^{(k)} \leq \kappa$), or $F_i = \emptyset$ otherwise. Solve for optimal \vec{d} in:

$$\hat{V}_t^{(k)} = \max_{\vec{d}} \sum_{s \in F_i} P_{s,F}(\vec{d}) \left(r_t^{(k)} + d_s - \left[\bar{V}_{t+1}^{(k-1)}(\vec{x}_t^{(k)}) - \bar{V}_{t+1}^{(k-1)}(\vec{x}_t^{(k)} + \mathbf{1}_s) \right] \right) + \bar{V}_{t+1}^{(k-1)}(\vec{x}_t^{(k)}),$$
 - 9: Update value function parameters $\vec{\gamma}$, θ using $\hat{V}_t^{(k)}$ with a stochastic gradient step to define the new approximation $\bar{V}_t^{(k)}(\vec{x})$ for all \vec{x} .
 - 10: Simulate customer's decision under prices \vec{d} and available slots F_i , and accordingly define next state $\vec{x}_{t+1}^{(k)}$ and $y_{t+1}^{(k)}$.
 - 11: **end for**
 - 12: $k \leftarrow k + 1$
 - 13: **end while**
-

Scaling	Policy	#Deliv	TotalCost	MeanCost	MeanPrice	MeanValue	Total Profit \pm Stddev	Gap (%)
0.8	<i>VS</i>	2408	4942	2.05	3.01	32.62	80856 \pm 48.84	0
	<i>F4</i>	2334	4874	2.09	4	32.61	80581 \pm 52.33	-0.34*
	<i>F5</i>	2256	4811	2.13	5	32.62	80058 \pm 51.50	-0.99*
	<i>OC-0</i>	2503	4941	1.97	0.29	34.36	81790 \pm 45.86	1.16*
	<i>OC-C</i>	2506	4942	1.97	0.29	34.35	81884 \pm 44.85	1.27*
	<i>OC-R</i>	2342	4939	2.06	2.49	34.56	81934 \pm 49.73	1.33*
	<i>OC-CR</i>	2385	4974	2.09	2.27	34.58	82906 \pm 50.23	2.53*
0.9	<i>VS</i>	2652	5159	1.95	3.01	32.59	89236 \pm 48.37	0
	<i>F4</i>	2576	5088	1.98	4	32.6	89198 \pm 49.84	-0.04
	<i>F5</i>	2498	5020	2.01	5	32.61	88933 \pm 54.61	-0.34*
	<i>OC-0</i>	2820	5327	1.89	-0.7	34.19	89098 \pm 40.97	-0.15*
	<i>OC-C</i>	2738	5154	1.88	0.13	34.33	89207 \pm 43.30	-0.03
	<i>OC-R</i>	2569	5127	2.00	2.66	34.55	90454 \pm 49.85	1.36*
	<i>OC-CR</i>	2608	5179	1.99	2.45	34.57	91373 \pm 49.08	2.39*
1	<i>VS</i>	2866	5349	1.87	3.01	32.55	96569 \pm 47.24	0
	<i>F4</i>	2793	5284	1.89	4	32.56	96810 \pm 48.69	0.25*
	<i>F5</i>	2714	5208	1.92	5	32.57	96772 \pm 51.04	0.21*
	<i>OC-0</i>	3015	5524	1.83	-0.8	34.15	95018 \pm 40.11	-1.61*
	<i>OC-C</i>	2936	5356	1.82	0.01	34.29	95356 \pm 40.38	-1.26*
	<i>OC-R</i>	2775	5356	1.93	2.73	34.57	98132 \pm 46.22	1.62*
	<i>OC-CR</i>	2802	5357	1.91	2.61	34.55	98745 \pm 46.29	2.25*
1.1	<i>VS</i>	3052	5534	1.81	3.01	32.52	102884 \pm 46.19	0
	<i>F4</i>	2982	5461	1.83	4	32.52	103470 \pm 47.36	0.57*
	<i>F5</i>	2907	5391	1.85	5	32.54	103728 \pm 47.74	0.82*
	<i>OC-0</i>	3177	5706	1.8	-0.88	34.11	99872 \pm 36.51	-2.93*
	<i>OC-C</i>	3100	5542	1.79	-0.09	34.27	100413 \pm 38.59	-2.4*
	<i>OC-R</i>	2937	5497	1.87	2.94	34.56	104634 \pm 47.01	1.70*
	<i>OC-CR</i>	2970	5521	1.86	2.76	34.53	105231 \pm 43.99	2.28*
1.2	<i>VS</i>	3212	5697	1.77	3.01	32.5	108346 \pm 41.50	0
	<i>F4</i>	3146	5627	1.79	4	32.5	109206 \pm 45.57	0.79*
	<i>F5</i>	3074	5550	1.81	5	32.52	109812 \pm 45.61	1.35*
	<i>OC-0</i>	3310	5880	1.78	-0.94	34.1	103867 \pm 35.78	-4.13*
	<i>OC-C</i>	3236	5714	1.77	-0.17	34.25	104557 \pm 36.66	-3.5*
	<i>OC-R</i>	3090	5661	1.83	3.06	34.54	110529 \pm 45.29	2.02*
	<i>OC-CR</i>	3112	5676	1.82	2.92	34.5	110751 \pm 41.82	2.22*

Table 2: “#Deliv” is the average number of deliveries under the respective policy, “TotalCost” the average total delivery cost, “MeanCost” is TotalCost/#Deliv, “MeanPrice” the average delivery charge, “MeanValue” the average order value in terms of profit before delivery costs, “TotalProfit” the average total profit after distribution, “StdDev” is the standard deviation of profits, and “Gap” is the percentage gap to the total profit achieved by policy *VS*. All percentage improvements that are statistically significant at the 95% level are indicated by an asterisk.